

BÁO CÁO ĐỒ ÁN CUỐI KỲ

Môn học : Máy học

Chủ đề báo cáo

Nhận diện sản phẩm bán lẻ

Trần Văn Truyền - 19522448

Lê Vinh Quang - 19522093

Nguyễn Hữu Hưng - 19521571

Giảng viên hướng dẫn :

Ths. Phạm Nguyễn Trường An

TS. Lê Đình Duy

I. Giới Thiệu

1.1. Mô tả bài toán

- Trong cuộc sống hiện đại, việc xếp 1 hàng dài để chờ thanh toán trong các siêu thị, trung tâm thương mại khiến người khó chịu và mất rất nhiều thời gian, làm cho các doanh nghiệp tổn thất rất nhiều. trong đó việc quét mã vạch là một trong những công đoạn tốn thời gian nhất. Đặc biệt trong thời gian dịch bệnh covid-19 đang hoành hành gần đây thì việc thanh toán chậm làm ùn tắc, người xếp hàng tập trung đông là rất nguy hiểm . Nhận



thấy những nguy hiểm và những rủi ro rất lớn nên nhóm của chúng em đã cùng nhau tìm những phương pháp để giải quyết.

- Chúng em nhận thấy được việc công nghệ nhận diện hình ảnh ngày càng phát triển và các dữ liệu hình ảnh sản phẩm nhiều thì nhóm chúng em xin đề xuất 1 giải pháp tăng tốc quá trình thanh toán như sau :
- Sử dụng camera hoặc điện thoại được gắn vào phía trên của quầy thanh toán, sau đó nhận diện hình ảnh của các món hàng trong một khu vực được quy định trong thời gian thực được liên kết với màn hình của quầy. Sau đó sẽ phân chia ra các món hàng nào nhận diện được món nào không nhận diện được.Đối với các món không nhận diện được vẫn sẽ tính tiền theo kiểu quét mã vạch
- Với giải pháp này của chúng em thì các cửa hàng, siêu thị sẽ giảm được đáng kể thời gian thanh toán, tránh được ùn tắc quầy thu ngân ở những khung giờ cao điểm. Cửa hàng sẽ tận dụng được tối đa các cơ sở vật chất sẵn có để làm tăng trải nghiệm người đi mua hàng

1.2. Input, Output

- **INPUT:** Là một bức ảnh từ trên chiểu xuống quầy thanh toán với ánh sáng trắng rõ ràng trong đó bao gồm nhiều món hàng không xếp chồng lên nhau trong một khu vực được định sẵn bằng một cái khay hay đường viền màu được chụp từ một camera hoặc điện thoại
- **OUTPUT:** Là bức ảnh từ input nhưng có các đường viền vuông bao quanh các món hàng có màu xanh, là món hàng nhận diện được và món và trên các thanh trên cùng được gán nhãn tên của món đồ.

1.3. Mô tả bộ dữ liệu

- Nhóm chúng em có tìm hiểu được một số bộ dữ liệu được thu thập sẵn có liên quan đến đồ án của chúng em như là :
 - + **SOIL-47**¹ : là tập dữ liệu sản phẩm tập trung vào việc thử nghiệm các thuật toán nhận dạng đối tượng dựa trên màu sắc, nó chứa 47 danh mục sản phẩm với 21 hình ảnh cho mỗi danh mục, được chụp từ 20 chế độ xem khác nhau, Hai bộ những hình ảnh như vậy được chụp trong các điều kiện ánh sáng khác nhau để các thuật toán kiểm tra yêu cầu cường độ chiếu sáng bất biến



Hình 2. Một số hình ảnh từ bộ dữ liệu Soil-47

¹ "SOIL-47: Surrey Object Image Library." 4 thg 6. 2001,
<http://www.ee.surrey.ac.uk/CVSSP/demos/colour/soil47/>

- + **Grozi- 120²:** là tập dữ liệu được đề xuất cho các cửa hàng tạp hóa nhận biết trong môi trường tự nhiên. Nó chứa 120 tạp hóa danh mục sản phẩm. Đối với mỗi loại sản phẩm, có 2 loại hình ảnh, một loại hình ảnh được thu thập từ web, loại hình ảnh khác được thu thập bên trong một cửa hàng tạp hóa. Tổng cộng, 11.870 hình ảnh được thu thập với 676 từ web và 11.194 từ cửa hàng.

1		2		3		4		5	
6		7		8		9		10	
11		12		13		14		15	
16		17		18		19		20	
21		22		23		24		25	
26		27		28		29		30	
31		32		33		34		35	

Hình 3. một số sản phẩm trong tập dữ liệu Grozi-120

- Nhận thấy các tập dữ liệu có sẵn chủ yếu phần lớn là các sản phẩm của nước ngoài , mẫu mã khác biệt nên không thể áp dụng được cho các cửa hàng , siêu thị ở Việt Nam. Vì vậy nhóm quyết định chỉ tham khảo các bộ dữ liệu và tự đi thu thập một bộ dữ liệu riêng biệt
- Các sản phẩm nhóm dự định thu thập là các sản phẩm bán lẻ ở các cửa hàng tạp hóa, siêu thị, thân quen với mọi người như là : nước ngọt các loại, các loại bánh kẹo, các loại đồ mỹ phẩm, các vật dụng cá nhân,...
- **Khó Khăn :** Nhưng do tình hình dịch bệnh covid-19 diễn biến, các vùng của các thành viên trong nhóm đa số đều đang thực hiện chỉ thị số 16 nên việc ra ngoài để thu thập data là việc cực kì khó khăn. Nên bộ dữ liệu mà

² "GroZi-120 Database." <http://grozi.calit2.net/grozi.html>.

nhóm thu thập được chủ yếu là các sản phẩm có sẵn trong nhà mỗi thành viên dẫn đến bộ dữ liệu bị hạn chế.

II. Các Nghiên Cứu Trước

2.1. Bài báo “Deep Learning for Retail Product Recognition: Challenges and Techniques³”

- Tác giả : Yuchen Wei, Son Tran, Shuxiang Xu, Byeong Kang and Matthew Springer
- Họ sử dụng nhiều phương pháp trên nhiều bộ dữ liệu khác nhau, các phương pháp được sử dụng :
 - + Phương pháp cổ điển : Nhận dạng sản phẩm được thực hiện bằng cách trích xuất các tính năng trên hình ảnh của bao bì.
 - + DeepLearning
 - + Convolutional Neural Networks
 - + Deep Learning for Object Detection
 - + Product Recognition Based on Deep Learning
- Các bộ dữ liệu họ đã sử dụng :
 - + GroZi-120
 - + DS2 dataset
 - + RPC dataset
 - + Cigarette Dataset
 - + Grocery Store Dataset
 - + GroZi-3.2k
- Kết quả đạt được :
 - + Với tập dữ liệu RPC

³ "Deep Learning for Retail Product Recognition: Challenges and"
<https://www.hindawi.com/journals/cin/2020/8875910/>

<i>Clutter mode</i>	<i>Methods</i>	<i>cAcc</i> (\uparrow)	<i>ACD</i> (\downarrow)	<i>mCCD</i> (\downarrow)	<i>mIoU</i> (\uparrow)	<i>mAP50</i> (\uparrow)	<i>mmAP</i> (\uparrow)
Easy	Single	0.02%	7.83	1.09	4.36%	3.65%	2.04%
	Syn	18.49%	2.58	0.37	69.33%	81.51%	56.39%
	Render	63.19%	0.72	0.11	90.64%	96.21%	77.65%
	Syn+Render	73.17%	0.49	0.07	93.66%	97.34%	79.01%
Medium	Single	0.00%	19.77	1.67	3.96%	2.06%	1.11%
	Syn	6.54%	4.33	0.37	68.61%	79.72%	51.75%
	Render	43.02%	1.24	0.11	90.64%	95.83%	72.53%
	Syn+Render	54.69%	0.90	0.08	92.95%	96.56%	73.24%
Hard	Single	0.00%	22.61	1.33	2.06%	0.97%	0.55%
	Syn	2.91%	5.94	0.34	70.25%	80.98%	53.11%
	Render	31.01%	1.77	0.10	90.41%	95.18%	71.56%
	Syn+Render	42.48%	1.28	0.07	93.06%	96.45%	72.72%
Averaged	Single	0.01%	12.84	1.06	2.14%	1.83%	1.01%
	Syn	9.27%	4.27	0.35	69.65%	80.66%	53.08%
	Render	45.60%	1.25	0.10	90.58%	95.50%	72.76%
	Syn+Render	56.68%	0.89	0.07	93.19%	96.57%	73.83%

Experimental results of the ACO task on RPC dataset

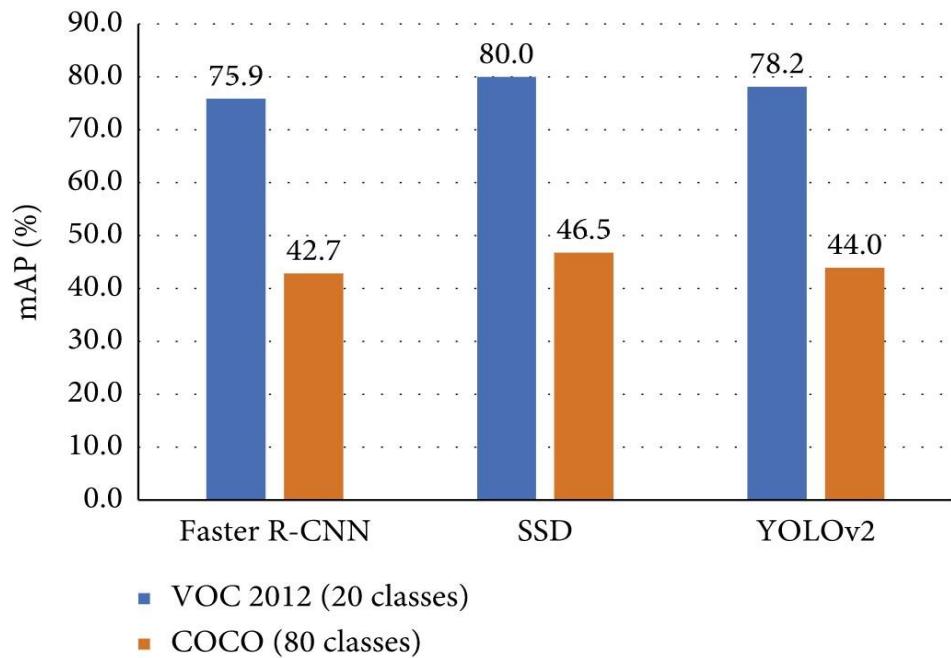
- + Với tập dữ liệu D2S

Approaches	mAP
Mask R-CNN	78.3
FCIS	68.3
Faster R-CNN	78.0
RetinaNet	80.1

2.2. Những thách thức của bài toán

- **Phân loại quy mô lớn** : Số lượng các sản phẩm riêng biệt cần được xác định trong siêu thị có thể rất lớn, khoảng vài nghìn, đối với một cửa hàng tạp hóa quy mô vừa vượt xa khả năng thông thường của máy dò đối tượng
- Hiện tại, YOLO, SSD, Faster R-CNN và Mask R-CNN là các phương pháp phát hiện đối tượng hiện đại, đánh giá thuật toán của chúng bằng PASCAL Bộ dữ liệu VOC và MS COCO . Tuy nhiên, PASCAL VOC chỉ chứa 20 lớp đối tượng và MS COCO chứa ảnh của 80 loại đối tượng.

Điều này có nghĩa là các thiết bị phát hiện đối tượng hiện tại không thích hợp để áp dụng trực tiếp vào nhận dạng sản phẩm bán lẻ do những hạn chế của chúng đối với các danh mục có quy mô lớn. Dưới đây là bảng so sánh kết quả trên bộ kiểm tra VOC 2012 (20 loại đối tượng) và COCO (80 loại đối tượng) với các thuật toán khác nhau, bao gồm Faster R-CNN, SSD và YOLOv2.



- **Giới hạn dữ liệu :** Các phương pháp tiếp cận dựa trên học sâu đòi hỏi một lượng lớn dữ liệu được chú thích để đào tạo, đặt ra một thách thức đáng kể trong những trường hợp chỉ có một số lượng nhỏ các ví dụ.
- Các công cụ gán nhãn hình ảnh, yêu cầu lao động thủ công để gắn nhãn mọi đối tượng trong mỗi hình ảnh. Thông thường, có ít nhất hàng chục nghìn hình ảnh đào tạo trong một tập dữ liệu phát hiện đối tượng chung, rõ ràng cho thấy rằng việc tạo một tập dữ liệu với đủ dữ liệu đào tạo cho học sâu là công việc tốn nhiều thời gian.
- **Các sản phẩm có tính chất thủy tinh :** Do sự tương đồng trực quan về hình dạng, màu sắc, văn bản và kích thước hệ mét giữa các

sản phẩm nội thủy tinh, các sản phẩm bán lẻ thực sự khó được xác định, việc máy tính phân loại các sản phẩm nội thủy tinh này sẽ rất phức tạp.

- **Tính linh hoạt :** Nhìn chung, với số lượng sản phẩm mới ngày càng nhiều, các cửa hàng tạp hóa cần thường xuyên nhập các mặt hàng mới để thu hút khách hàng. Hơn nữa, sự xuất hiện của các sản phẩm hiện có thường xuyên thay đổi theo thời gian. Do những lý do trên, một hệ thống ghi nhận thực tế nên linh hoạt mà không cần hoặc không phải đào tạo lại bất cứ khi nào một sản phẩm / gói sản phẩm mới được giới thiệu . Tuy nhiên, mạng nơ-ron tích tụ luôn bị “catastrophic forgetting” , chúng không thể nhận ra một số đối tượng đã học trước đó khi thích nghi với một nhiệm vụ mới.



Hình 5. Các hãng liên tục thay đổi ngoài sản phẩm và ra mắt nhiều loại mới

III. Xây Dựng Bộ Dữ Liệu

• 3.1 Tại sao cần thu thập dữ liệu thủ công:

- Do các dữ liệu có sẵn không đáp ứng được ngữ cảnh của bài toán như: background xung quanh sản phẩm, sản phẩm nội địa Việt Nam,
- Việc tự thu thập dữ liệu giúp em kiểm soát được các yếu tố ngoại cảnh như góc quay, sản phẩm, ánh sáng, ... tùy vào đó mà đưa ra các tiêu chí để thu thập dữ liệu

● 3.2 Các tiêu chí thu thập dữ liệu:

- Với ngũ cảnh bài toán là chụp một bức ảnh từ trên xuống trong môi trường siêu thị ánh sáng rõ ràng vì vậy yêu cầu đặt ra là có gắng mô phỏng được khoảng cách từ camera đến sản phẩm, ánh sáng ,góc quay cụ thể như sau:

- + Vị trí camera điện thoại cách sản phẩm từ 20-30 cm
- + Các vật phải nằm trong một khu vực giới hạn (đối với tập test)
- + Background màu trắng
- + Các sản phẩm có nhiều tư thế có thì phải lấy được tất cả các tư thế của vật
- + Ưu tiên lấy các mặt có màu sắc logo rõ ràng
- + Đảm bảo điều kiện ánh sáng trong phòng tốt mô phỏng càng giống ánh sáng trong siêu thị càng tốt
- + Độ phân giải tối thiểu là 1280x720



Hình 6: Ảnh được chụp từ khoảng cách 30cm có background trắng, ánh sáng rõ ràng, với 2 góc chụp khác nhau và 2 tư thế khác nhau của vật

● 3.3 Cách thức thu thập:

- Chuẩn bị các thiết bị:

- + Một cái bàn xoay tự chế với các linh kiện được mua từ shopee⁴
- + Một cái giá treo điện thoại
- + Một chiếc điện thoại chụp ảnh có độ phân giải tối thiểu 720p
- + Một nguồn sáng trong phòng tốt



Hình 7: Ảnh minh họa cho bàn xoay

- Cách thức thu thập tập dữ liệu train:

- + Chuẩn bị ánh sáng, góc quay, tư thế của vật
- + Đặt vật lên bàn xoay rồi bắt bàn xoay
- + Quay video có độ dài tùy vào tốc độ xoay (10-30s)
- + Sau đó thay đổi tư thế của vật và góc quay rồi tiếp tục quay
- + Một video có độ dài từ 10-30s tùy vào tốc độ xoay sẽ được cắt ra thành 30 frame ảnh cho vào bộ dữ liệu

- Cách thức thu thập dữ liệu test:

- + Chuẩn bị một background trắng giới hạn 30x30cm, góc quay từ trên xuống, ánh sáng phòng tốt
- + Sau đó bỏ 2-6 vật không xếp chồng lên nhau vào một hình rồi chụp lại

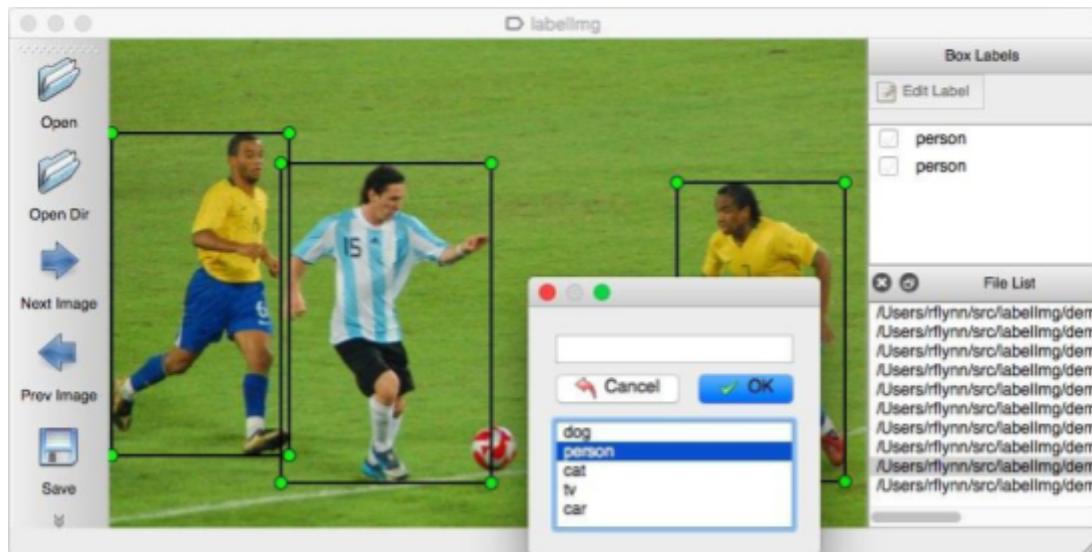
⁴ "How To Make a Turning Table Product | Viet Toys - YouTube." <https://www.youtube.com/watch?v=G12m2QF13Co>.



Hình 8: Một số ảnh tử tập test

• 3.4 Gán nhãn dữ liệu:

- + Sau tổng hợp được dữ liệu thô thì tụi em bắt đầu gán nhãn, tụi em sử dụng tool LabelImg⁵

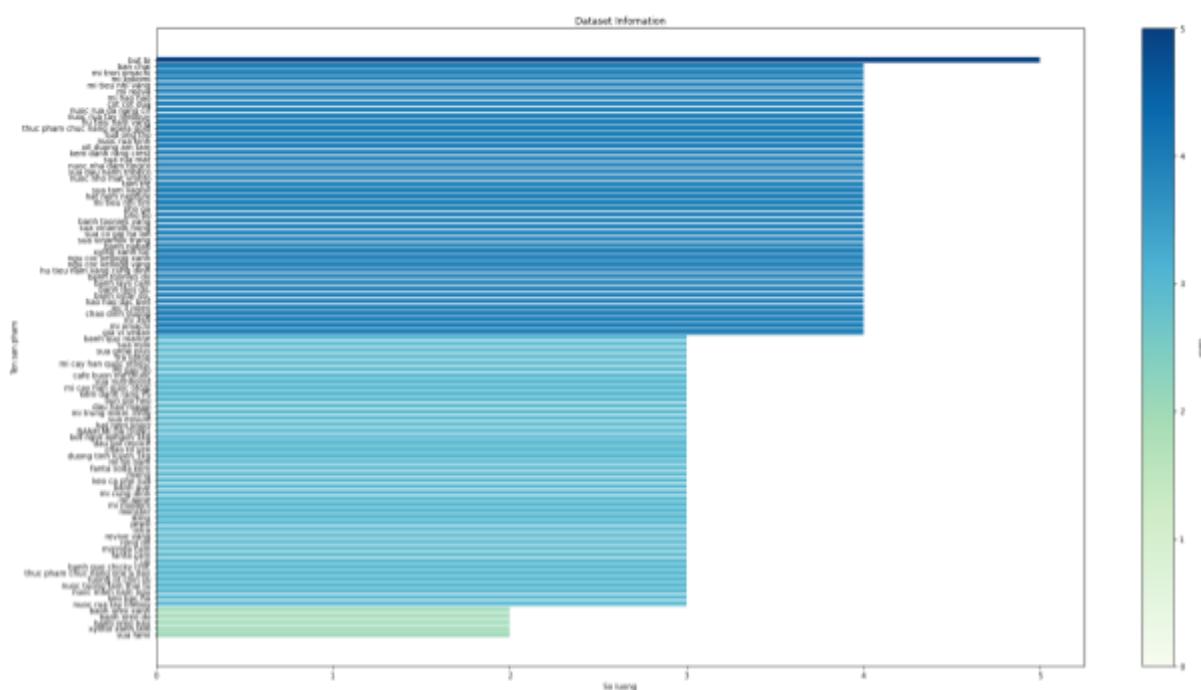


Hình 9: Tool label

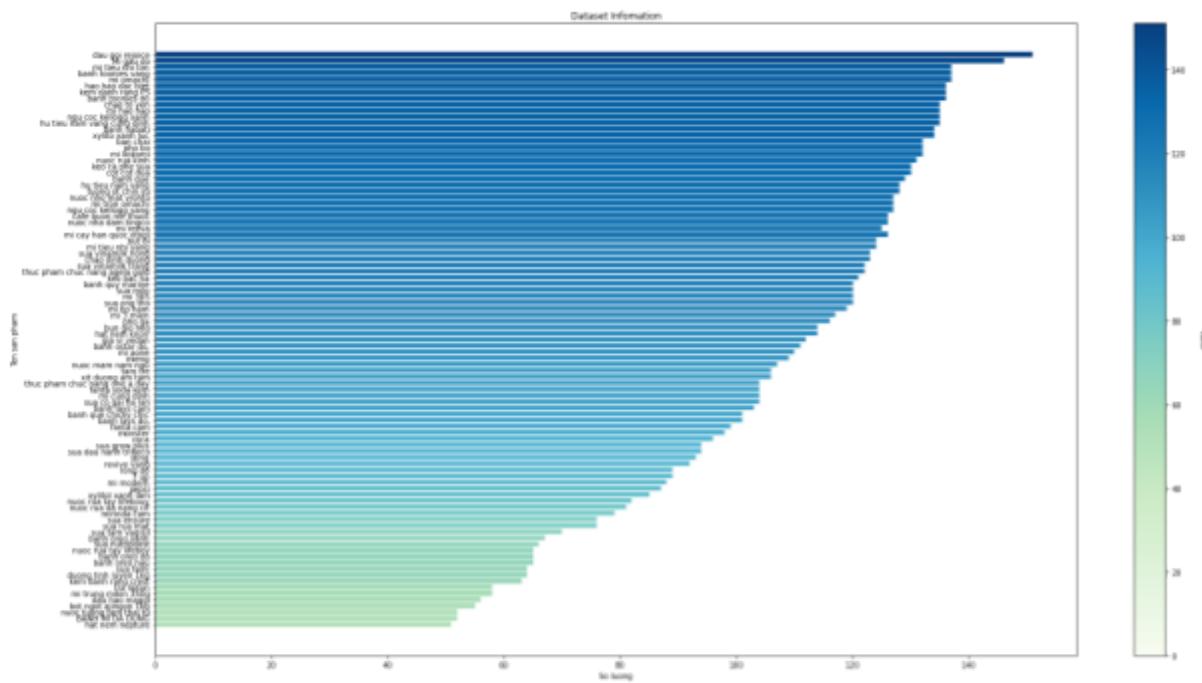
⁵ "LabelImg is a graphical image annotation tool and label ... - GitHub." <https://github.com/tzutalin/labelImg>.

• 3.5 Tổng quan bộ dữ liệu:

- Bộ dữ liệu gồm 2 tập dữ liệu là tập train và tập test của 94 class là các sản phẩm tiêu dùng trong siêu thị như: mì, bánh kẹo, nước ngọt, đồ dùng cá nhân,...
 - Bộ dữ liệu có 323 video có độ dài 10-30s với trung bình hơn 3 video cho một vật, số lượng video nhiều nhất của một vật là 5 còn thấp nhất là 2
 - Bộ dữ liệu dùng để train bao gồm: 9974 ảnh chỉ có một vật được cắt ra từ video của 94 class cụ thể:
 - Số video nhiều nhất của một vật là
 - Cụ thể số ảnh nhiều nhất của một vật là 151 còn thấp nhất là 52, trung bình mỗi vật có 106 ảnh
 - Mỗi ảnh chỉ có một vật vậy có tổng cộng 9974 vật được gán nhãn

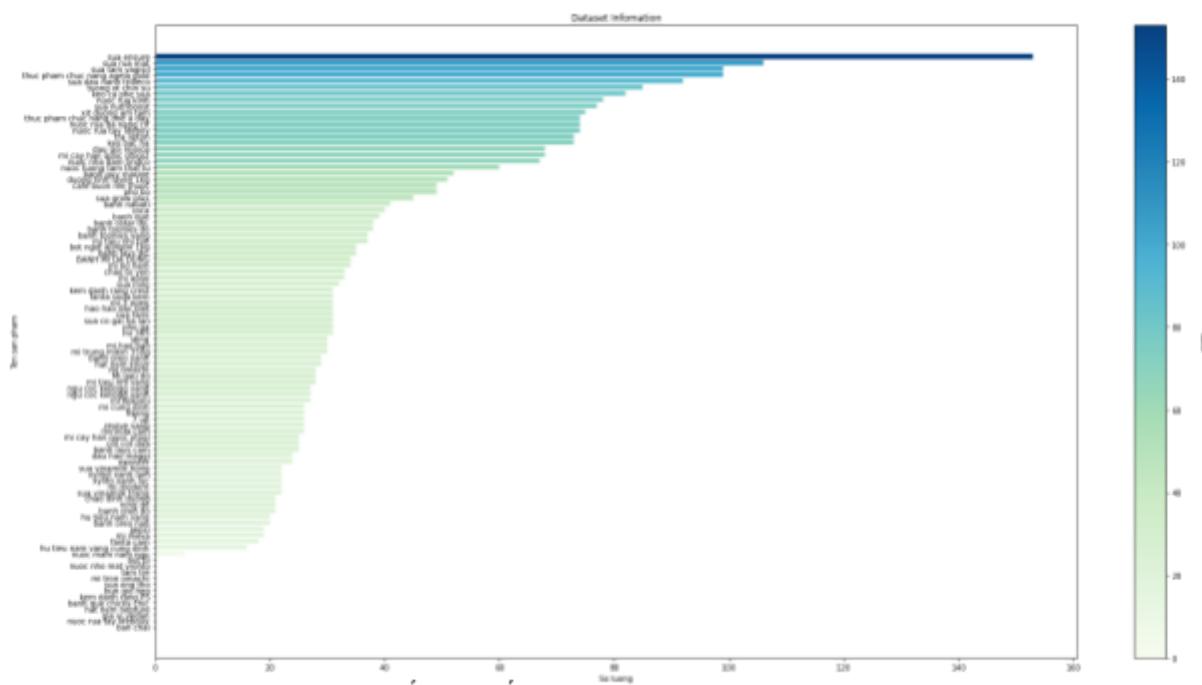


Hình 10 : Bảng thống kê số video



Hình 11: Thống kê số lượng ảnh trong tập train

- Bộ dữ liệu test bao gồm: 939 ảnh tự chụp từ 2-6 vật trong mỗi ảnh
 - Trong đó có 3433 vật đã được gán nhãn, trung bình mỗi ảnh có 3-4 vật trong một tấm ảnh, ảnh nhiều nhất của một vật là 153, có 12 vật không có ảnh nào



Hình 12: Thống kê số lượng vật được gán nhãn trong tập test



Hình 13: Bộ dữ liệu đa dạng về hình dạng



Hình 14: Các vật có hình dạng giống nhau



Hình 15: vật có hình dạng giống nhau nhưng khác màu sắc

Nhận xét:

- + Bộ dữ liệu do tui em thu thập thủ công nên khá sạch, số lượng class trung bình không quá lớn chỉ ngang với các nghiên cứu trước (94 so với 120 của Grozi- 120 và 200 của RPC)
- + Bộ dữ liệu train vẫn còn chênh lệch (151 ảnh gấp 3 lần 52 ảnh cho ảnh nhiều và ít nhất)
- + Bộ ảnh test còn tệ hơn vậy khi vật lớn nhất có 153 và có tận 12 vật không có ảnh nào

=> Chính vì các yếu tố trên tui em quyết định sẽ tăng cường thêm dữ liệu

- **Tăng cường dữ liệu:**

- Nhận thấy bộ dữ liệu train và đặc biệt là test đang chênh lệch rất nhiều đồng thời có sự gợi ý của thầy nên tui em quyết định tăng cường dữ liệu

- **Cách thức tăng cường dữ liệu⁶:**

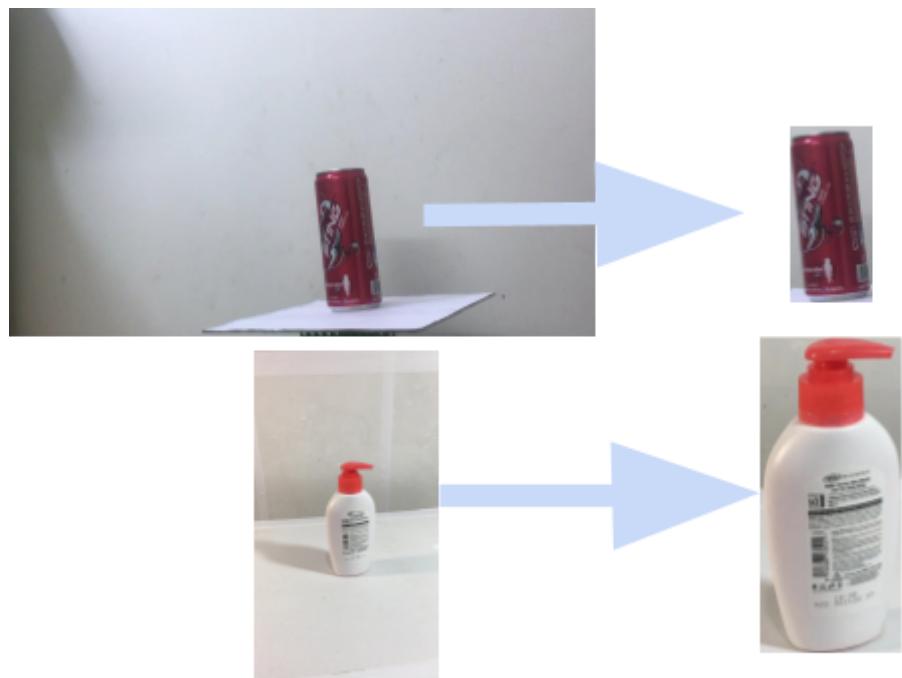
- + Đầu tiên tui em sẽ cắt các bounding box từ ảnh của train ra
- + Sau đó ghép ngẫu nhiên 3-6 vật vào một tấm ảnh
- + Trước đó để đảm bảo thêm được nhiều ảnh của các vật có ít ảnh và thêm ít lại các ảnh đã có nhiều tui em lấy ngẫu nhiên:

$$50 + 153 - sl_i$$

- + Trong đó: 50 là số lượng ảnh được lấy ngẫu nhiên từ mỗi vật, 153 là số lượng ảnh lớn nhất của một vật trong tập train, sl_i là số lượng của lần lượt các vật trong tập train trước khi tăng cường

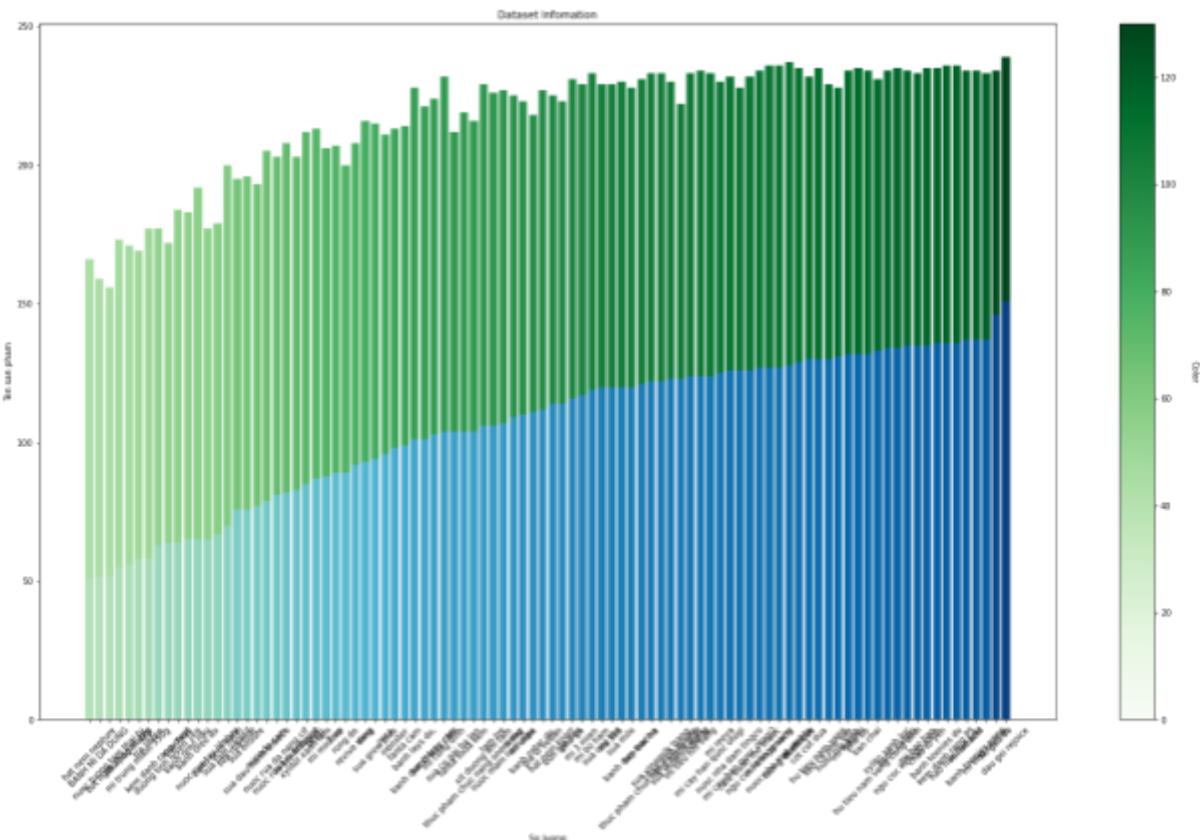
⁶ "Mapping an image with random coordinates, using PIL, without them" 2 thg 2. 2019,
<https://stackoverflow.com/questions/54488217/mapping-an-image-with-random-coordinates-using-pil-without-them-stay-one-on-to>.

⁷ "felixchenfy/Data-Augment-and-Train-Yolo - GitHub."
<https://github.com/felixchenfy/Data-Augment-and-Train-Yolo>.



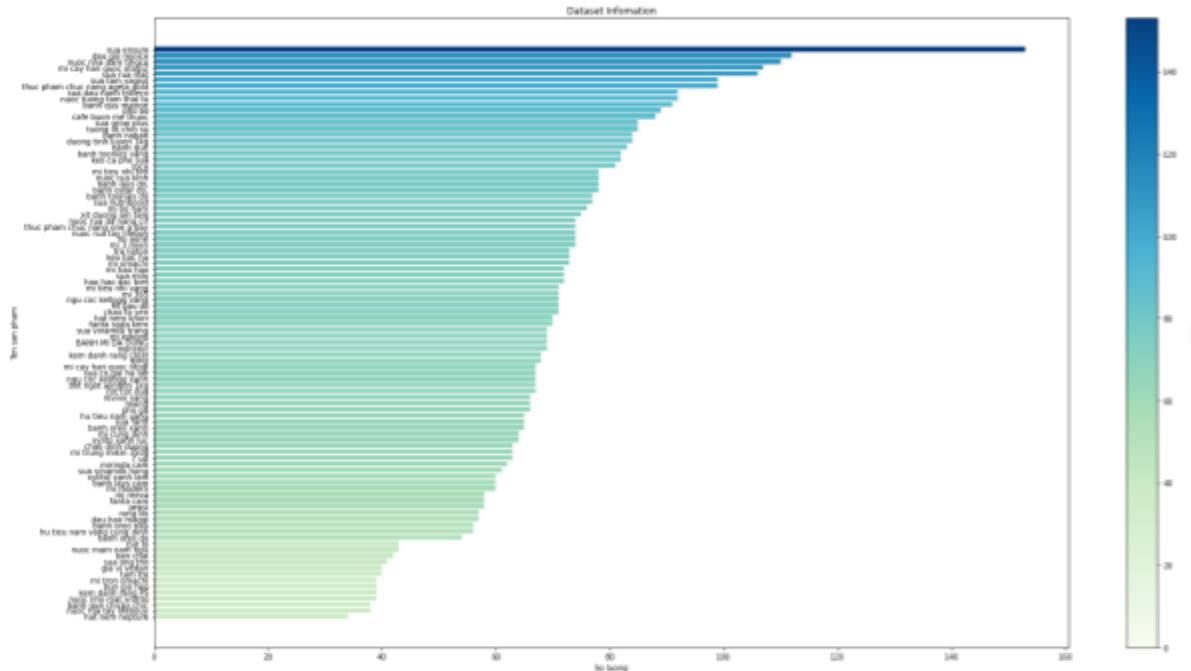
Hình 16: Ảnh được cắt bounding box ra

- + Tổng cộng bọn em có thêm được 10422 vật được chia ngẫu nhiên vào 2320 tấm ảnh với trung bình 4.5 vật/ảnh trong tập train



Hình 17: Thống kê số lượng vật sau khi tăng cường tập train

- + Tổng cộng bọn em có thêm được 3117 vật được chia ngẫu nhiên vào 687 tấm ảnh với trung bình 4.5 vật/ảnh trong tập train



Hình 18: Thống kê số lượng vật trong test set sau khi tăng cường



Hình 19: Minh họa tập train sau khi tăng cường

IV. Training Và Đánh Giá Model

4.1 Phương thức đánh giá model

- Để đánh giá model Object detection người ta sử dụng các thông số như IoU, AP, mAP,
- IoU là độ overlap giữa Ground-truth-bounding box là đường bao mà ta gán nhãn với Predicted bounding box là đường bao mà model dự đoán

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

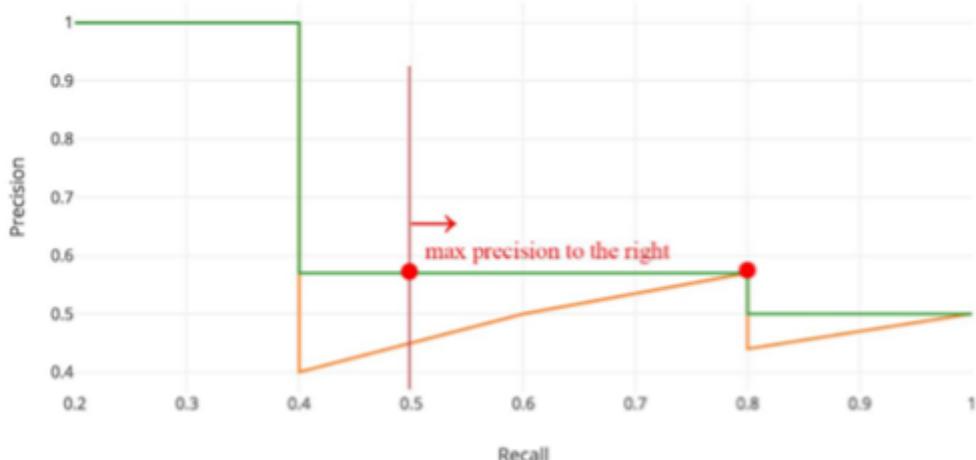

Hình 20: Công thức tính IoU

- Precision: đo lường mức độ chính xác là dự đoán của mô hình tức là tỷ lệ phần trăm dự đoán của mô hình là chính xác.
- “Recall” đo lường như thế nào tốt mô hình tìm thấy tất cả các mẫu tích cực
- AP: là chỉ số có quan hệ mật thiết với chỉ số Precision (phần trăm bbox được dự đoán đúng) và Recall (tỉ lệ phần trăm các bbox được đoán đều chính xác)
- AP: là độ chính xác với IoU = 0.5
- AP75: là độ chính xác với IoU = 0.75

$$Precision = \frac{TP}{TP + FP} = \frac{\text{Số dự đoán chính xác}}{\text{Tổng số lần dự đoán}}$$

$$Recall = \frac{TP}{TP + FN} = \frac{\text{Số lần dự đoán chính xác}}{\text{Số lần nhận dạng đúng có thể có}}$$

Hình 21: Cách tính Precision và Recall

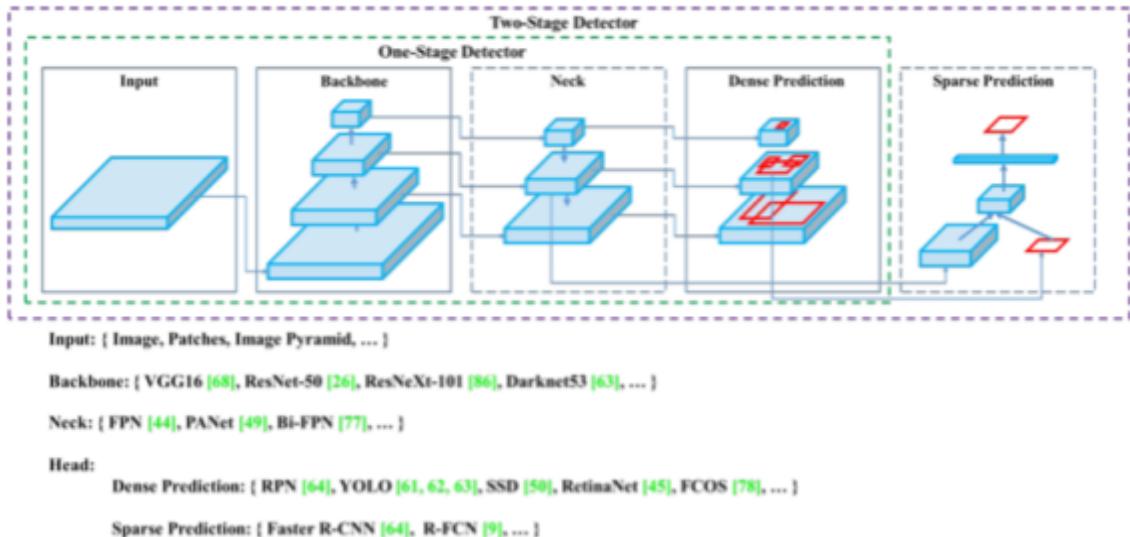


Hình 22: Cách tính AP dựa trên Precision và Recall

- Chỉ số mAP là trung bình tổng chỉ số AP của tất cả các class

4.2 Yolov4

- Yolov4 được đánh giá là model để xây dựng object detector tốt nhất
- Model Yolov4 sử dụng từ nhiều bộ dataset để train từ trước, đơn cử nhất là từ hai bộ dataset nổi tiếng là ImageNet(ILSVRC 2012 val) gồm 1000 object classes với gần 1,5 triệu ảnh dùng để huấn luyện và MS COCO (test-dev 2017) gồm 80 classes với 330000 ảnh dùng để huấn luyện, có thêm các bước tăng cường dữ liệu như cutmix, blur,...
- Sử dụng kiến trúc backbone CPSDarknet53 (Kết hợp Darknet-53 và chiến lược CPSNet) để trích xuất đặc trưng. Sau khi các đặc trưng được trích xuất dưới dạng output là một feature map, nó sẽ được đưa vào các layers để dự đoán labels cũng như bbox của vật thể.

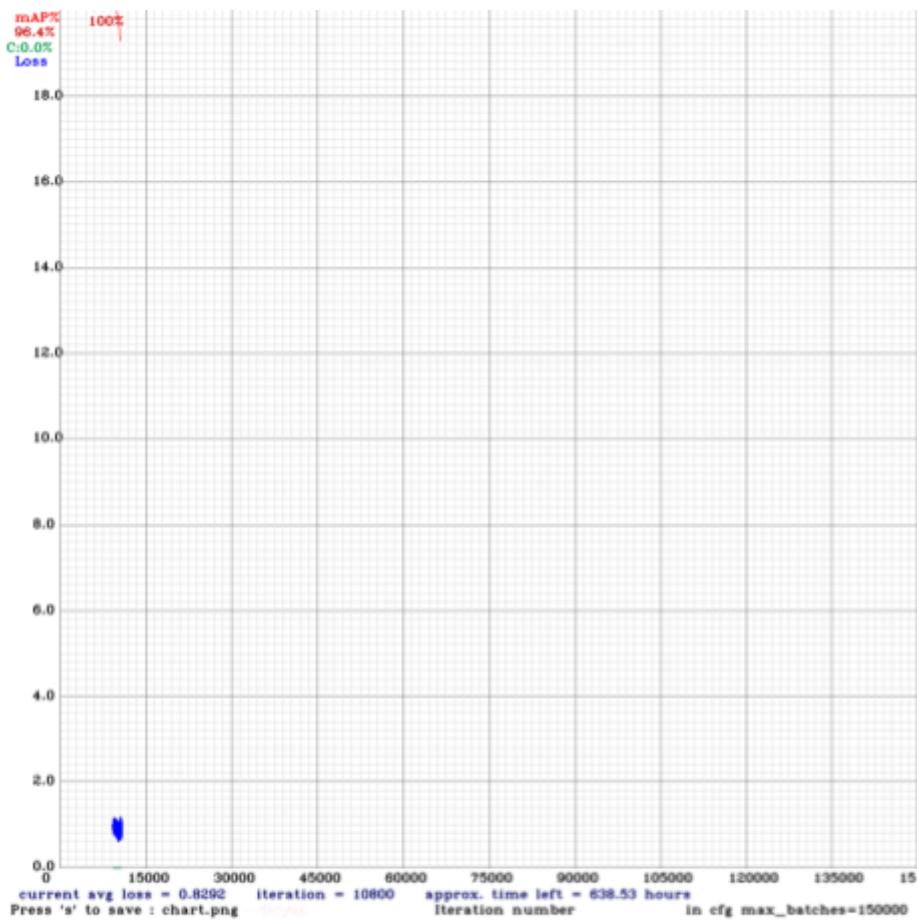


Hình 23: Kiến trúc của yolov4

- Tui em dùng Pretrained Model với yolov4.conv.137⁸ để train lại model nhằm tiết kiệm thời gian train thay vì train lại từ đầu:
- Chuẩn bị dữ liệu và training cho Model yolov4⁹:
 - + Gồm một folder chứa 12294 ảnh và 12294 folder
 - + File yolo.names danh sách tất cả các class
 - + File Makefile dùng để bật GPU cho model train
 - + File yolov4-custom.cfg tùy chỉnh để training nhanh nhất có thể
- Do thời gian train trên tập data là khá lâu nên chung em chỉ train 10k iteration với 8k iteration đầu là chỉ có dữ liệu thường và 2k iteration sau là được train với dữ liệu tăng cường

⁸ "Alexey AlexeyAB - GitHub." <https://github.com/AlexeyAB>.

⁹ "Train Yolov4 trên Colab, chi tiết và đầy đủ từ A-Z - Mí AI." 25 thg 5. 2020,
<https://www.miai.vn/2020/05/25/yolo-series-train-yolo-v4-train-tren-colab-chi-tiet-va-day-du-a-z/>



Hình 24: Kết quả tập trên tập val được lấy từ tập train

- Kết quả được train trên tập test như sau:

- + Trước khi tăng cường:

mAP trên tập val = 99%

mAP trên tập test = 64.33%

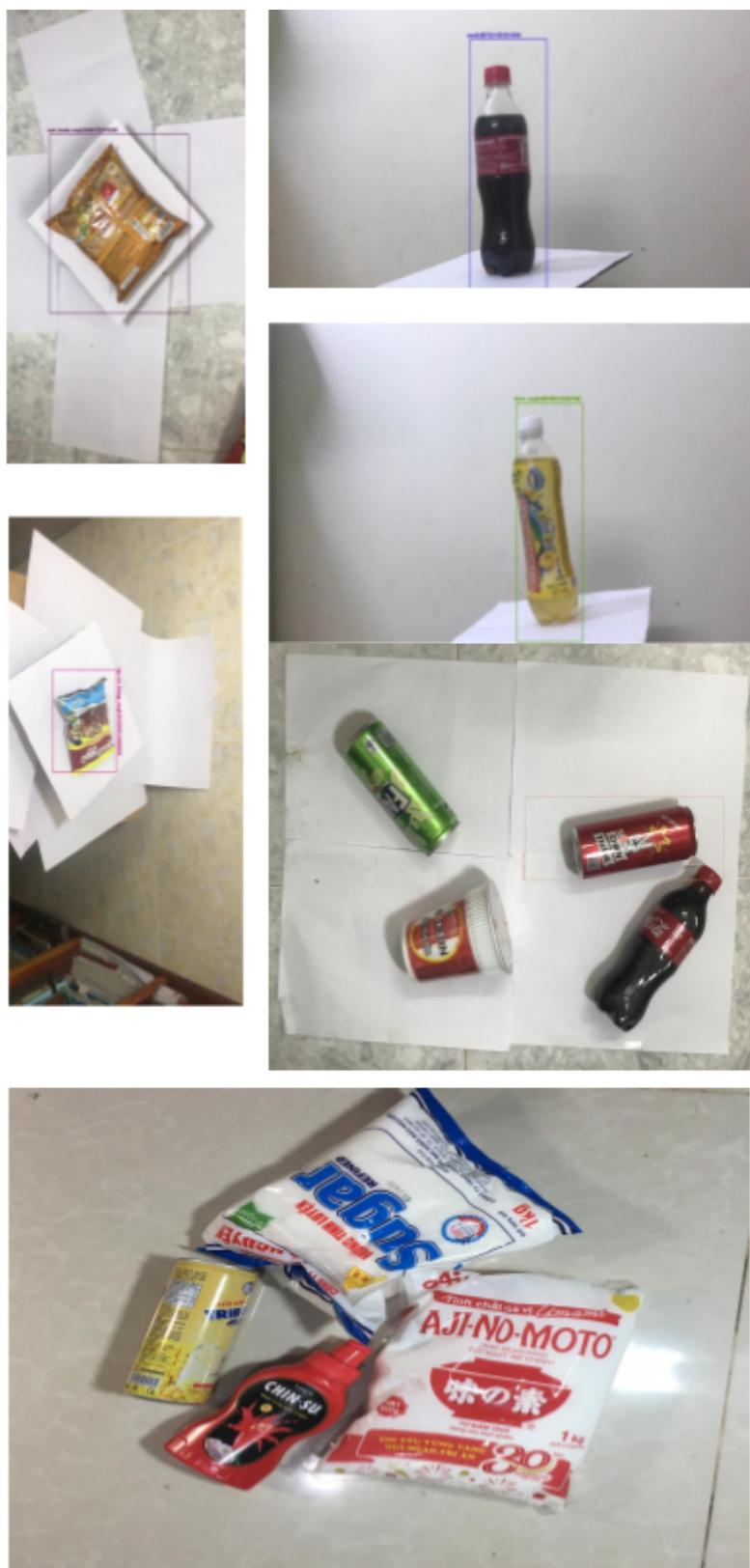
=> Model bị overfitting khá nặng và tỉ lệ nhận diện cũng không hề cao, và sau khi test lại một số hình như bên dưới thì có kết luận rằng model có khả năng nhận biết nhiều vật trong một ảnh không cao

- + Sau khi tăng cường

mAP trên tập val = 99%

mAP trên tập test = 78.61%

=> Kết quả được cải thiện thấy rõ tần khoang 14% khá ổn so với các hình trước đó chỉ thấp hơn Retina Net với 80% ngang ngửa với Faster R-CNN và Mask R-CNN, đồng thời vẫn còn hơi overfitting với tập val 99%

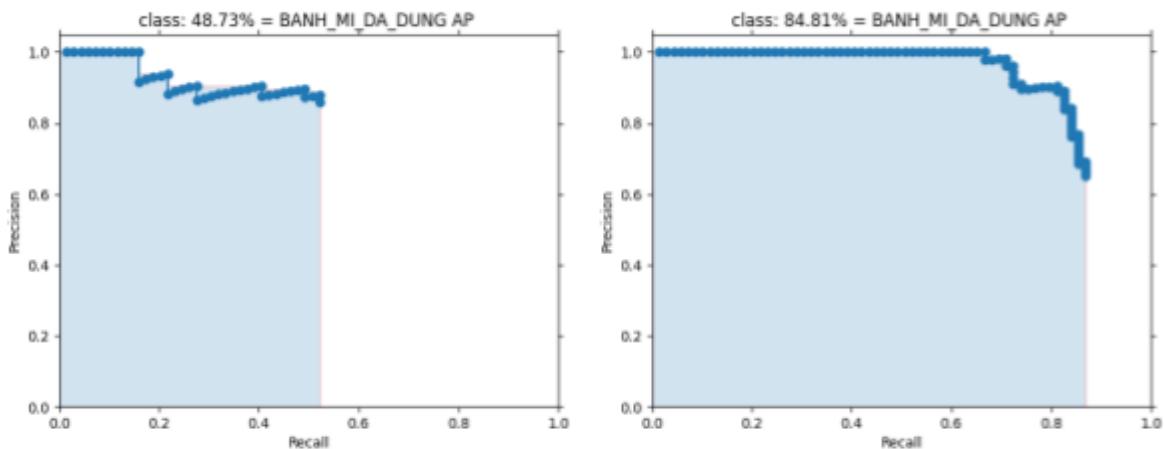


Hình 25: Ảnh detect vật trước khi tăng cường dữ liệu

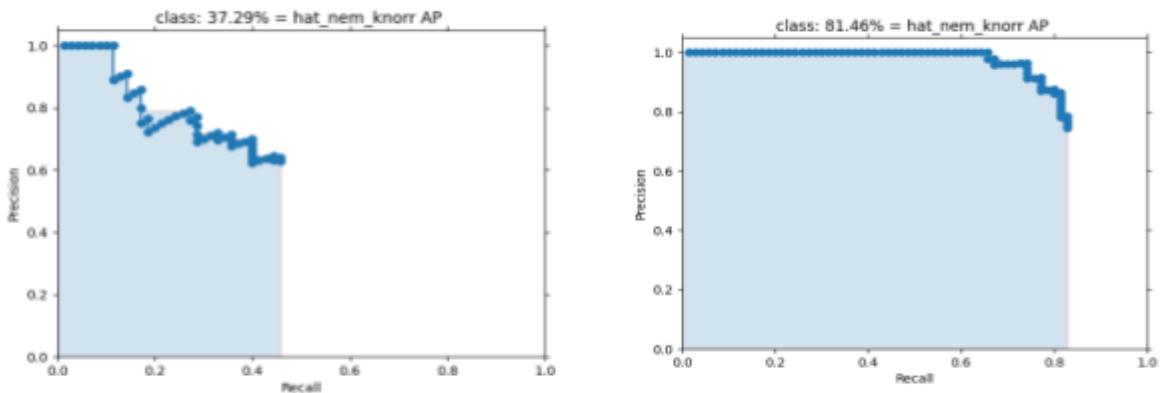


Hình 26: chỉ số mAP trên tập test của trước và sau khi tăng cường

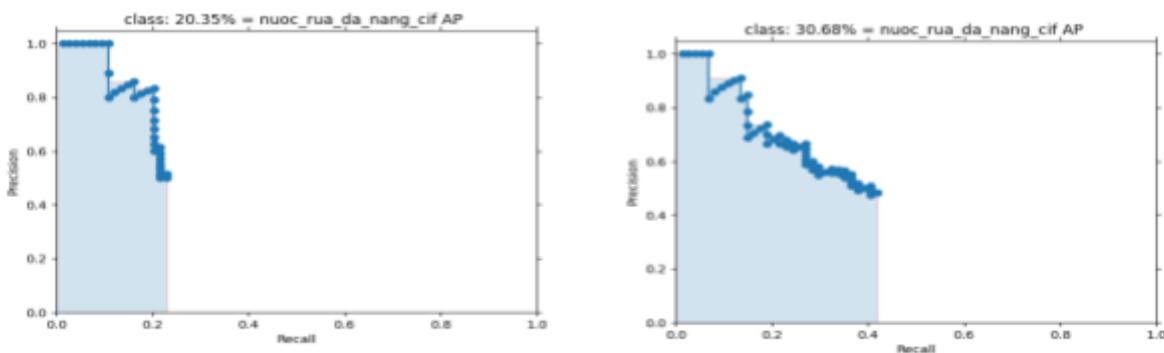
- Từ đây ta dễ dàng thấy được mAP của các vật được cải thiện tuy nhiên một số vật có mAP thấp cải thiện không nhiều như



Hình 27: Chỉ số AP50 của class BANH_MI_DA_DUNG trước và sau khi tăng cường



Hình 28: Chỉ số AP50 của class Hat_nem_knorr trước và sau khi tăng cường



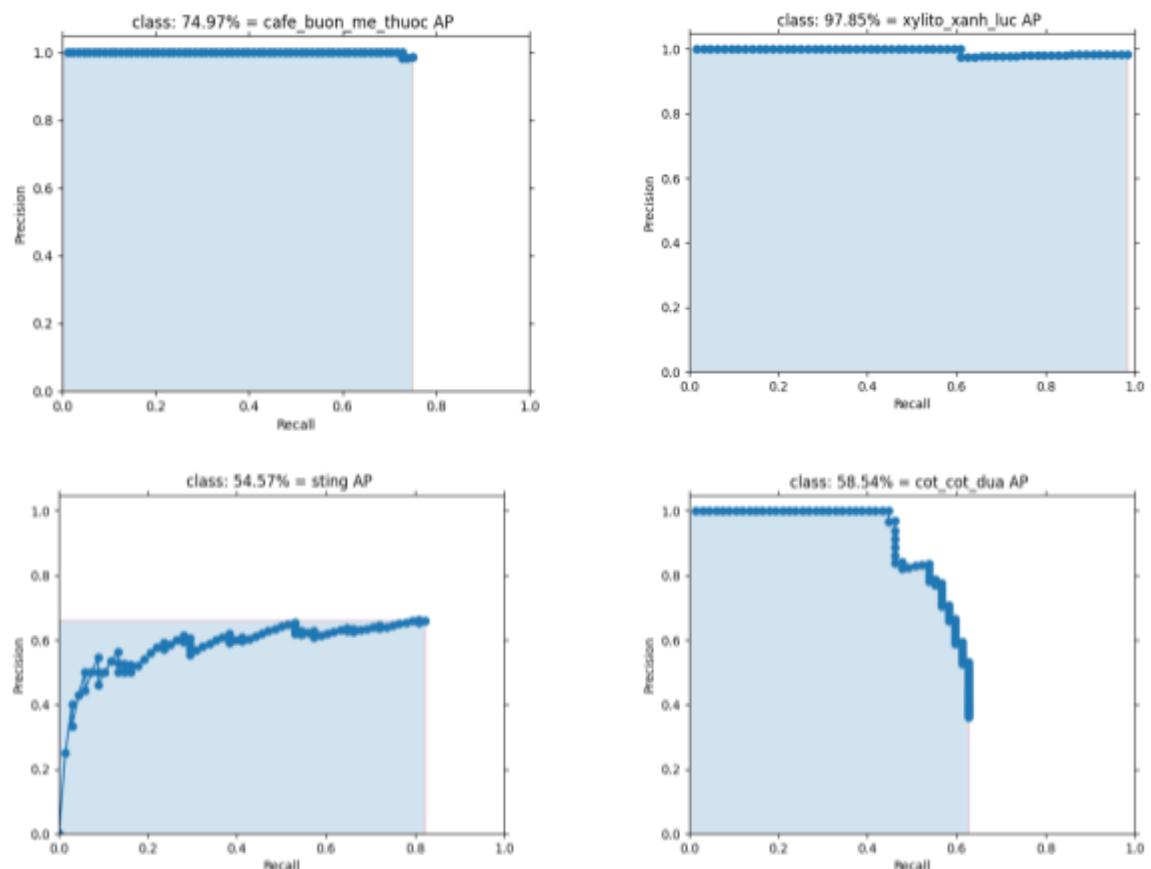
Hình 29: Chỉ số AP50 của class nuoc_ra_da_nang không cải thiện được nhiều



Hình 30: Ảnh detect trước và sau khi tăng cường dữ liệu



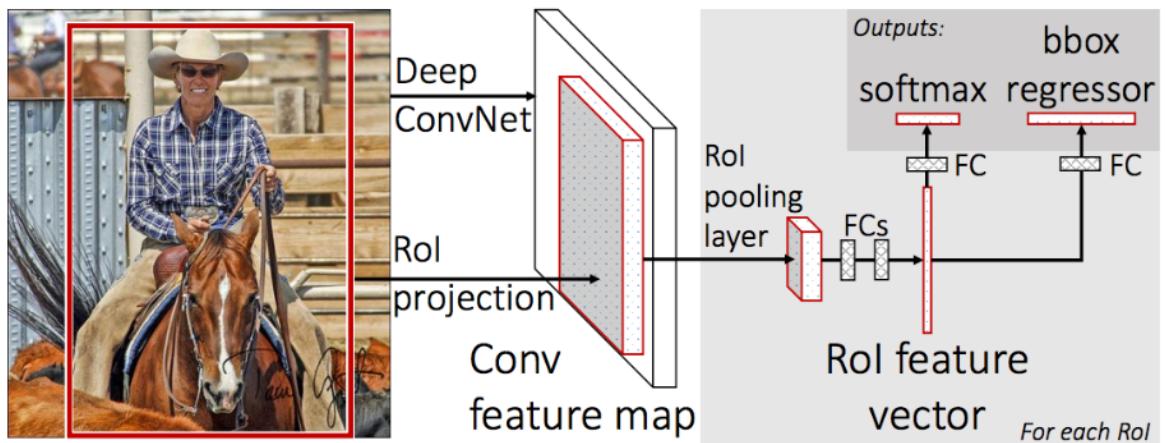
Hình 31: Ảnh nhận diện sau khi tăng cường



Hình 32: AP50 của một số sản phẩm sau khi tăng cường

4.3 Faster-rcnn

- Trong bài toán này, chúng em sử dụng mô hình Faster R-CNN trên Detectron2 do Facebook Reseach xây dựng.
- Detectron2 được xây dựng bởi Facebook, một thư viện cung cấp mã nguồn mở để training model trên custom dataset của nhóm. Model zoo của Detectron2 rất phong phú, có thể sử dụng để pretrained và chúng em quyết định lựa chọn model Faster R-CNN để pretrained.
- Detectron2 là một thư viện mô hình thị giác máy tính mô-đun dựa trên PyTorch phổ biến. Đây là lần lặp lại thứ hai của Detectron, ban đầu được viết bằng Caffe2. Hệ thống Detectron2 cho phép cắm các công nghệ thị giác máy tính hiện đại tùy chỉnh vào quy trình làm việc .
- Detectron2 bao gồm tất cả các mô hình có sẵn trong Detectron ban đầu, chẳng hạn như Faster R-CNN, Mask R-CNN, RetinaNet và DensePose. Nó cũng có một số mô hình mới, bao gồm Cascade R-CNN, Panoptic FPN và TensorMask, và chúng tôi sẽ tiếp tục bổ sung thêm các thuật toán khác. Chúng tôi cũng đã thêm các tính năng như Định mức hàng loạt đồng bộ và hỗ trợ cho các tập dữ liệu mới như LVIS
- Faster R-CNN là một mô hình với cấu trúc gần như tương tự với R-CNN và được cải thiện đáng kể về tốc độ training, sau khi trích xuất đặc trưng ảnh Faster R-CNN không sử dụng thuật toán để tìm ra khu vực có khả năng chứa các đối tượng mà thêm hẳn một mạng CNN để tìm ra nó.



- Kiến trúc single model Fast R-CNN (được trích xuất từ bài báo gốc). Ở bước đầu ta áp dụng một mạng Deep CNN để trích xuất ra feature map. Thay vì warp image của region proposal như ở R-CNN chúng ta xác định ngay vị trí hình chiếu của region proposal trên feature map thông qua phép chiếu ROI projection. Vị trí này sẽ tương đối với vị trí trên ảnh gốc. Sau đó tiếp tục truyền output qua các layer ROI pooling layer và các Fully Connected layers để thu được ROI feature vector. Sau đó kết quả đầu ra sẽ được chia làm 2 nhánh. 1 Nhánh giúp xác định phân phối xác suất theo các class của 1 vùng quan tâm ROI thông qua hàm softmax và nhánh còn xác định tọa độ của bounding box thông qua hồi qui các offsets.
- Mô hình này nhanh hơn đáng kể cả về huấn luyện và dự đoán, tuy nhiên vẫn cần một tập hợp các region proposal được đề xuất cùng với mỗi hình ảnh đầu vào.

Quá trình chuẩn bị dữ liệu và training

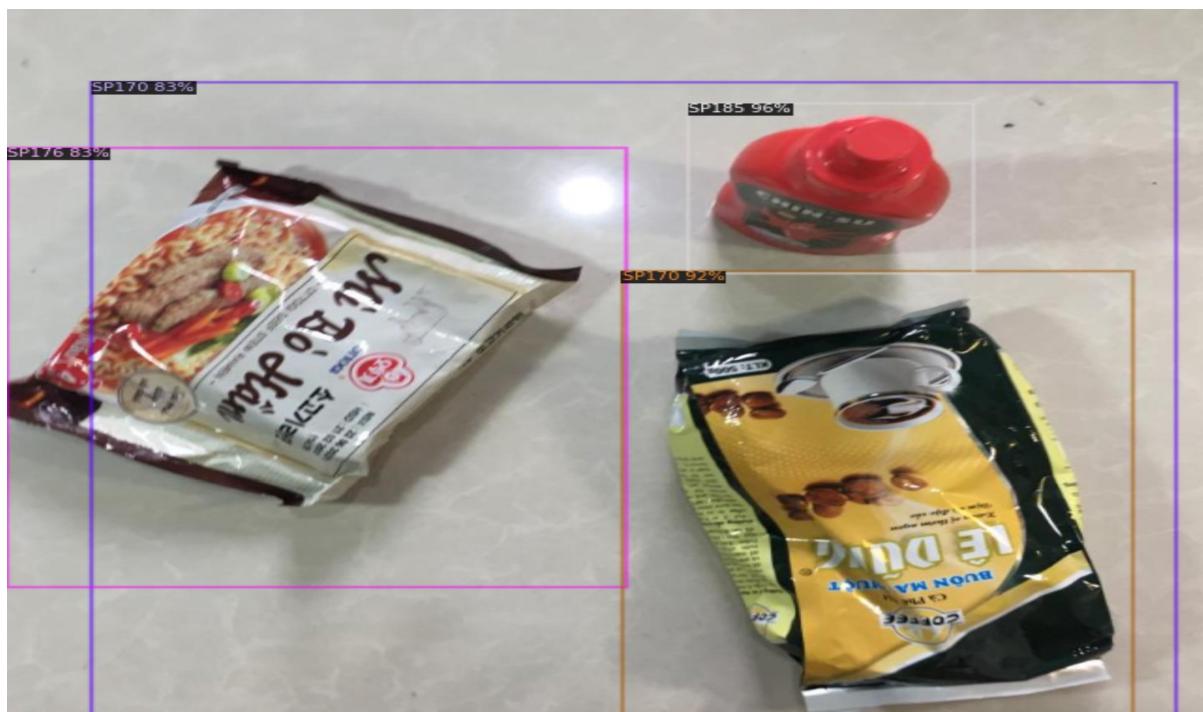
- Roboflow có hỗ trợ để chuẩn bị các chuyển đổi dữ liệu từ file format VOC *.xml sang file format COCO *.json, chỉ cần up các file .jpg, .xml, .txt lên website của họ, data sẽ tự động được convert sang file format COCO .json, sau đó lưu lại link và lên Colab tải về.

- Quá trình training : training theo các code mẫu sẵn đã được hướng dẫn trên <https://www.youtube.com/watch?v=4OXntFVfFio>, các thông số đều để mặc định, chỉ tăng “MAX_ITERS” lên mỗi lần tải và train tiếp model
 - + IMS_PER_BATCH = 4
 - + CLASSES = 200 (= 199 + 1)
 - + EVAL_PERIOD = 500
 - + BATCH_SIZE_PER_IMAGE = 64
- Các khó khăn trong quá trình train model : Quá trình Training diễn ra nhanh hơn so với model yolov4, tuy nhiên vẫn còn lâu và chậm, chuẩn bị file tốn thời gian, khi muốn train phải tải lại các data từ link roboflow, rồi đăng ký các tập train, test, và khi muốn test trên tập test khác phải đăng ký trên một tên khác. Quá trình chuẩn bị phức tạp. Và do nhóm bắt đầu train khá trễ nên model có độ chính xác chưa cao.
- Nguyên nhân: số lượng data nhiều, nhưng với kích thước data lớn thì cũng không thực sự khả quan và tụi em liên tục gặp lỗi CUDA out of memory trong quá trình train.
- Đây là link colab dùng để train model detectron 2 của nhóm :

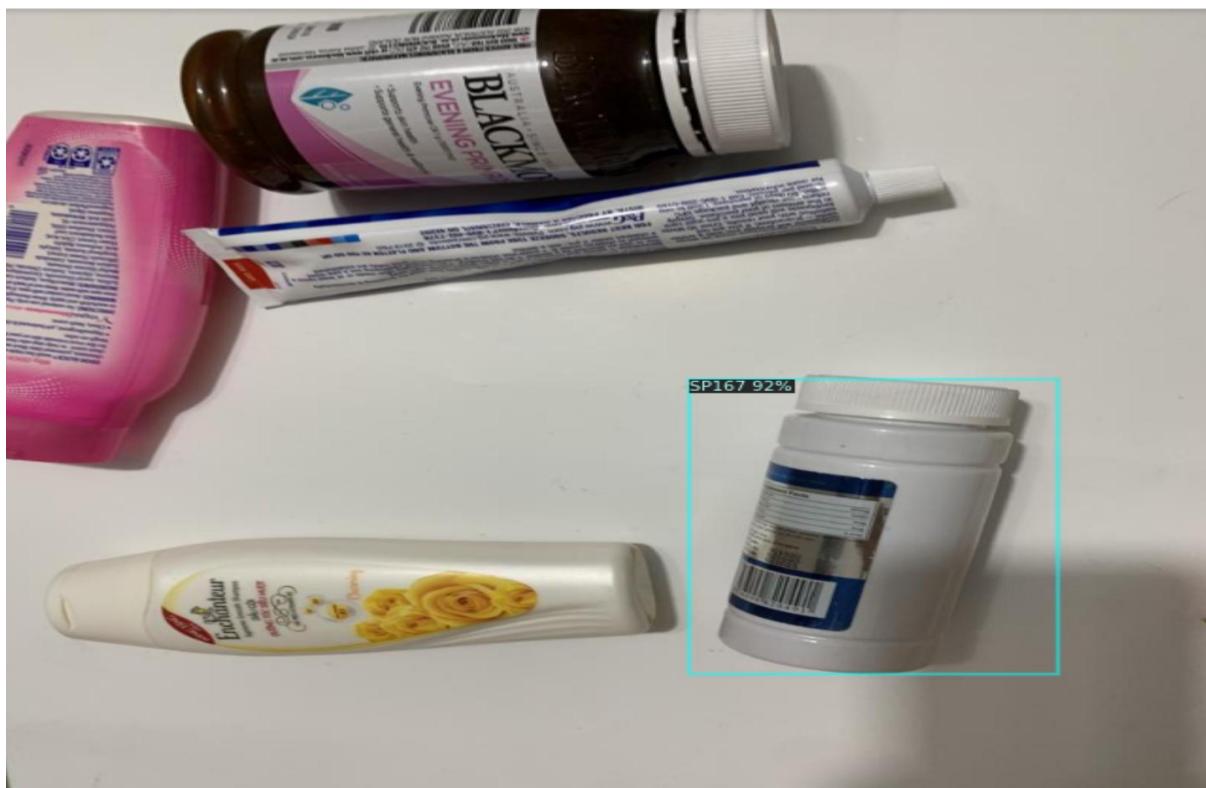
https://colab.research.google.com/drive/126qYqSbjd4H87Vm_QaoFgTlf1hDnOIpQ?usp=sharing

Đánh giá Model

- Model Faster R-CNN sử dụng framework detectron2 sử dụng các chỉ số mAP, AP50, AP75 để đánh giá độ dự đoán chính xác của model.
- Kết quả từ tập val được lấy từ tập train khi train tới **6000 iteration** thì **AP50** đạt khoảng : **90,5%**
- Kết quả trên tập test : **mAP = 67,15 %**
- Kết quả thấp nguyên do là vì model train chưa đủ lâu và chỉ có 1500 iteration được train với dữ liệu tăng cường nên kết quả không tốt như yolov4
- Một số hình ảnh khi test :



- Trường hợp model nhận diện dư một sản phẩm , 4 trong khi trong ảnh chỉ có 3 object



- Trường hợp này model nhận diện khá tệ chỉ phát hiện được 1 trên tổng số 5 sản phẩm , nguyên do bởi vì các sản phẩm này được xếp lại rất gần, hầu như không có khoảng trống giữa các sản phẩm, và các sản phẩm có màu sắc gần như tương tự nhau



- Các trường hợp mà model nhận diện được tốt khi các sản phẩm được xếp tách rời nhau, màu sắc giữa các sản phẩm có sự khác biệt rõ rệt



V. Ứng dụng và hướng phát triển

- Hiện nay qua việc khảo sát nhóm nhận thấy việc khách hàng chờ đợi thanh toán trong các cửa hàng tiện lợi và siêu thị vào các khung giờ cao điểm tốn rất nhiều thời gian, bên cạnh đó dịch covid19 đang diễn biến rất phức tạp thì việc giảm thiểu thời gian thanh toán càng được chú trọng.
- Đầu tư về thiết bị để thu thập dữ liệu nên tạo một không gian riêng để dành cho chụp hình các sản phẩm để tăng khả năng nhận biết sản phẩm lúc thanh toán một cách tốt nhất.
- Liên kết với ngân hàng hoặc ví điện tử để thanh toán và nhận hóa đơn hàng ngay trên điện thoại, hạn chế nhận hóa đơn giấy như hiện nay.

VI. Tài liệu tham khảo

- Model Yolov4:

[YOLO Series] Train YOLO v4 train trên COLAB chi tiết và đầy đủ (A-Z)

<https://www.miai.vn/2020/05/25/yolo-series-train-yolo-v4-train-tren-colab-chi-tiet-va-day-du-a-z/>

YOLO You Only Look Once

<https://phamdinhkhanh.github.io/2020/03/09/DarknetAlgorithm.html>

Tìm hiểu mô hình YOLO cho phát hiện vật - Từ YOLOv1 đến YOLOv5

<https://aicurious.io/posts/tim-hieu-yolo-cho-phat-hien-vat-tu-v1-den-v5/>

Darknet

<https://github.com/AlexeyAB/darknet>

YOLO - object detection <https://opencv-tutorial.readthedocs.io/en/latest/yolo/yolo.html>

- Model Faster R-CNN:

How to Use the Detectron2 Model Zoo (for Object Detection)

<https://blog.roboflow.com/how-to-use-the-detectron2-object-detection-model-zoo>

Detectron2 Model Zoo and Baselines

https://github.com/facebookresearch/detectron2/blob/master/MODEL_ZOO.md#faster-r-cnn

Triển khai Faster RCNN cho các bài toán detection

<https://viblo.asia/p/trien-khai-faster-rcnn-chocac-bai-toan-detection-OeVKBMoE5kW>

How to Train a Custom Faster R-CNN Model with Facebook AI's Detectron2 | Use Your Own Dataset <https://www.youtube.com/watch?v=4OXntFVfFio>

- Dánh giá Model:

mAP (mean Average Precision) might confuse you!

<https://towardsdatascience.com/map-mean-average-precision-might-confuse-you-5956f1bfa9e2>

mAP (mean Average Precision)

<https://dothanhblog.wordpress.com/2020/04/24/map-mean-average-precision/>

Series YOLOv4: #3 Dánh giá model bằng mAP -Object detection

<https://devai.info/2020/12/17/tim-hieu-mapmean-average-precision-danh-gia-mo-hinh-object-detection-su-dung-yolov4/>